Współczesne moduły automatycznego sprawdzania pisowni dla języka polskiego i angielskiego

Tomasz Jach

Zakopane, 2008 r.

1 Wstęp

W pracy przedstawiono proces sprawdzania poprawności słów dla języka polskiego (będącego przykładem języka fleksyjnego, a co za tym idzie - trudnego do automatycznej analizy słownikowej) oraz angielskiego. Przybliża się również największe problemy związane z tym procesem, omawia sposób działania programu Hunspell (wykorzystywanego m.in w Firefoxie, Thunderbirdzie i OpenOffice'ie), budowę i sposób generowania list słów oraz wzorców odmiany (z dokonaniem porównania ilościowego złożoności tychże plików dla języka angielskiego i polskiego).

W podsumowaniu nakreśla się możliwość wykorzystywania Hunspella do analizy semantycznej tekstu, docelowo - do automatycznej analizy dokumentów pod kątem grupowania na podstawie automatycznie generowanych słów kluczowych.

1.1 Różnice w budowie, składni i fleksji języków: polskiego i angielskiego

Język polski należy do grupy języków zachodniosłowiańskich. Jego alfabet składa się z 32 liter. Posiada one dwie liczby (pojedynczą i mnogą) oraz wymierającą i nieklasyfikowaną już liczbę podwójną (np. ręka - rękoma). W gramatyce naszego języka ojczystego wyróżniamy aż 5 rodzajów: tradycyjnie wymieniany rodzaj żeński, nijaki oraz 3 z grupy rodzajów męskich: męski osobowy, męski nieosobowy żywotny, męski nieosobowy nieżywotny.

W języku polskim występuje 7 różnych przypadków. Jest to i tak skromna liczba w porównaniu do języka węgierskiego - będącego przedstawicielem klasy języków zlepkowych - mającego ich 29. Ubogi w tym zakresie jest język angielski - należący do klasy języków pozycyjnych - gdzie występują tylko dwa przypadki, do tego zwykle rozróżniane tylko dla zaimków.

Każdy z przypadków charakteryzuje się złożonymi regułami zastosowywania, do których istnieją liczne wyjątki.

Ze względu na czasy oraz aspekty wyróżniamy dwie grupy czasowników: czasowniki w formie niedokonanej (przyszłej złożonej, teraźniejszej, przeszłej niedokonanej) oraz w formie dokonanej (przyszłej prostej, przeszłej dokonanej). Jako wart odnotowania archaizm występuje jeszcze czas zaprzeszły. Wyróżnia się również trzy tryby czasowników (oznajmujący, przypuszczający, rozkazujący) oraz trzy strony (bierną, czynną, zwrotną). Oprócz powyższych, istnieje wiele rodzajów imiesłowów, każdy odmieniany i traktowany w oddzielny sposób.

Wszystkie powyższe informacje są jedynie skrótowym zarysem, lecz w sposób doskonały obrazują złożoność naszego języka w porównaniu np. do języka angielskiego. W odróżnieniu od języka polskiego, w angielskim praktycznie zanikła fleksja. Cała deklinacja ogranicza się do szczątkowych przypadków, które

są sukcesywnie wypierane. Odmiana przez przypadki dotyczy jedynie zaimków, rzeczowniki posiadają jedynie mianownik oraz dopełniacz. Język angielski wywodzi się z grupy języków zachodniogermańskich. W alfabecie wyróżniamy 26 liter. Biorąc pod uwagę konstrukcje gramatyczne, w języku angielskim wyróżniamy 16 czasów. Nie jest to jednak kwestia kluczowa pod względem automatycznego sprawdzania pisowni. Podobnie jak w języku polskim, angielski posiada takie same trzy strony.

Jak wspomniano wcześniej, nie występuje tutaj fleksja, co z uwagi na ograniczone możliwości tworzenia poprawnych form pobocznych znacząco ułatwia proces automatycznego sprawdzania pisowni. Tendencją światową jest znaczne uproszczenie gramatyki oraz fleksji języka. Proces ten obecny jest również w naszym ojczystym języku, a widać go zwłaszcza w tekstach technicznych najeżonych zapożyczeniami, głównie z języka angielskiego.

1.2 Rozwiązania automatycznego sprawdzania pisowni

Obecnie występuje kilka równorzędnych mechanizmów sprawdzania pisowni. Pierwszym z nich jest **Ispell**.

Program ten powstał w latach 70' ubiegłego wieku za sprawą R.E. Goriona. Początkowo, napisany był w języku assemblera, dopiero jakiś czas później dokonano przeportowania na C. Jego cechą charakterystyczną była obsługa wyłącznie większości zachodnich języków oraz sugerowanie poprawnych wyrazów oddalonych jedynie o jeden, używając do pomiarów dystansu Damerau–Levenshteina (czyli wymagających tylko jednej operacji zamiany, wstawienia bądź usunięcia litery) [8]. Obecnie system ten jest bardzo rzadko spotykany za sprawą swoich ograniczeń.

Następcą Ispella był **Aspell**. Jest on obecnie domyślnym systemem sprawdzania pisowni w większości aplikacji Open Source. Głównym odpowiedzialnym za rozwój programu jest Kevin Atkinson. W odróżnieniu od Ispella, radzi sobie z tekstami zakodowanymi w standardzie UTF-8, a co za tym idzie - nie posiada ograniczenia do języków zachodnioeuropejskich. Dodatkowym atutem w porównaniu do poprzednika jest możliwość korzystania z wielu baz słownikowych jednocześnie [7].

Kolejnym programem, również wykorzystywanym do dzisiaj, jest **MySpell**. Wspierany był przez fundację Mozilla oraz zespół OpenOffice.org. Ideą przyświecającą powstaniu była integracja dotychczasowych rozwiązań sprawdzania pisowni w jeden efektywnie działający mechanizm. W programie tym pojawiły się słowniki przenoszenia wyrazów (z jęz. ang. *hyphenation*) oraz tezaurusy. Po raz pierwszy wprowadzono również koncepcję plików afiksowych oraz słownikowych w celu zaoszczędzenia miejsca i zwiększenia szybkości wyszukiwania.

Myspell został obecnie wyparty przez system **Hunspell**. Płynnie przejął on rolę poprzednika w programach takich jak OpenOffice oraz Mozilla Firefox. W systemie tym pojawiły się mechanizmy do analizy morfologicznej tekstu stworzone specjalnie dla języków o bogatej fleksji oraz dużej ilości złożeń i zrostów. Pierwotnie (co wskazuje nazwa) opracowywany był dla języka węgierskiego. System radzi sobie również z językami aglutynacyjnymi (czyli językami używającym afiksów określających funkcję składniową wyrazu w wypowiedzeniu). Niestety, do tej pory trwają prace nad przystosowaniem go do skomplikowanej gramatyki fińskiej.

2 Hunspell

Reguły rządzące poprawnością słów w danym języku w przeważającej większości przypadków dają się zapisać w formie zrozumiałej przez komputer. Dzięki znacznie szybszemu przetwarzaniu danych przez maszynę jest ona w stanie sprostać zadaniu sprawdzania poprawności pisowni w czasie rzeczywistym.

Niestety, w odróżnieniu od człowieka posługującego się danym językiem, maszyna cyfrowa nie posiada zmysłu "intuicji", lecz musi zostać zaprogramowana w celu użycia zapisanych reguł. Reguły te występują w słownikach niezbędnych do poprawnego działania każdego programu sprawdzania pisowni, w tym również omawianego **Hunspella**.

2.1 Budowa plików słownikowych

Kompletny słownik w programie Hunspell składa się z trzech odrębnych plików, z których każdy spełnia odrębną funkcję. Dwa pierwsze (słownik afiksowy oraz słownik słów) są obowiązkowe, słownik tezaurusowy jest opcjonalny.

Słownik słów

```
5
abadańczyk/NOqsT
abadański/bXxYc
Abaddon/O
abaka/MnN
abakanka/mMN
```

Słownik słów posiada tylko dwie sekcje. W pierwszej, będącej w rzeczywistości jedną linijką, należy podać przybliżoną liczbę pozycji występujących w słowniku. Ma to na celu przyspieszenie alokowania pamięci dla słownika.

W następnych linijkach aż do końca plików znajduje się lista poprawnych słów w języku. Definicję słowa kończy znak "/", po nim następują identyfikatory afiksów występujących w słowniku afiksowym. Identyfikatory te wskazują na możliwe metody tworzenia słów pochodnych ze słowa bazowego występującego w słowniku słów.

Słownik afiksowy Przykładowy fragment słownika słów wygląda następująco:

```
SET ISO8859-1
TRY
esianrtolcdugmphbyfvkwzESIANRTOLCDUGMPHBYFVKWZ'

PFX A Y 1
PFX A   0     re          .

SFX V N 2
SFX V   e     ive         e
SFX V   0     ive         [^e]

REP 1
REP a ei
```

Pierwsza sekcja określa rodzaj zastosowanego kodowania znaków, a następnie wyszczególnia wszystkie możliwe litery danego języka względem częstości występowania. Druga sekcja określa możliwe prefiksy oraz sufiksy w danym języku. Słowo kluczowe "PFX" oznacza prefiks, "SFX" - sufiks. Po nich następuje jednoznakowe oznaczenie afiksa (użyte w słowniku słów). We wszystkich słownikach rozróżniane są wielkie i małe litery. Kolejne pole określa, czy prefiks może występować wspólnie z innym sufiksem i odwrotnie. Oddzielona jedną spacją za symbolem "Y" lub "N" znajduje się długość definicji danego afiksa. Każdy afiks posiada tyle definicji, ile określono w jego nagłówku. Składają się one z trzech nowych

pól. Pierwsze określa jakie znaki należy usunąć przed dodaniem afiksu, aby utworzyć poprawne słowo. Następnie znajduje się sam afiks, który zostaje dołączony do słowa, a na końcu - za pomocą wyrażeń regularnych zapisany jest warunek występujących liter, konieczny do spełnienia przed zastosowaniem afiksu. Ostatnią sekcję wypełniają w całości "replikanty", czyli zestawy znaków, które mogą być używane wymiennie w procesie sprawdzania poprawności słowa.

Słownik synonimów składa się z dwóch plików. Pierwszy, z rozszerzeniem .idx ma na celu zwiększenie szybkości wyszukiwania. W każdej linii tego pliku znajduje się informacja o słowie ujętym w tezaurusie oraz odległości początku pliku ze słownikiem od definicji słowa wewnątrz niego. Przykładowo, wartość 16 oznacza, że należy przeskoczyć pierwsze 16 znaków pliku ze słownikiem, aby odnaleźć poszukiwaną definicję.

Plik .dic ma następujący wygląd:

```
a jakże|4
-|ajuści|ależ tak|ba
-|i owszem|jużci (przestarz.)
-|naturalnie|tak właśnie
-|jeszcze czego
```

Elementy definicji jednego słowa oddzielane są znakiem "|". W pierwszej linii obecne jest wyrażenie lub zwrot, do którego w kolejnych n liniach (gdzie n jest kolejnym polem), również oddzielone znakami "|" znajdują się zebrane synonimy.

2.2 Sposób działania programu Hunspell

Od czasów pierwszych edytorów tekstu i pierwszych mechanizmów sprawdzania pisowni dokonał się niebywały postęp jeśli chodzi o sprawność i wydajność algorytmów. Z drugiej strony - współczesne języki ewoluują w dużo szybszym tempie niż to miało miejsce jeszcze kilkadziesiąt lat temu.

Obecnie programy takie jak Hunspell nie tylko sprawdzają, czy słowo jest poprawne, ale także w przypadku gdy słowo uważane jest za niepoprawne, sugerują jego poprawne brzmienie. Co więcej - dostarczane są również narzędzia do przeprowadzania analizy morfologicznej tekstu.

- **Sprawdzanie poprawności pisowni**

  Główną funkcjonalnością programu Hunspell jest automatyczne sprawdzanie poprawności pisowni. Program w wersji konsolowej (a taka docelowo może zostać wykorzystana w automatycznej klasyfikacji) generuje następujące wyniki. Przykładowym tekstem jest fragment pracy dyplomowej z zakresu dataminingu.

  ```
  Hunspell 1.2.6
  Wiedza odkryta w procesie dataminingu
  jest wykorzystywana w praktyce do analizy
  i poprawy jakości różnych czynników.
  *
  + odkryć
  *
  + proces
  & dataminingu 4 26: glutaminianu,
  glutaminian, witaminizując,
  witaminizacja
  *
  + wykorzystywać
  ```

```
*
+ praktyka
*
+ analiza
*
+ poprawa
+ jakość
+ różny
+ czynnik
```

Jak widać, program w przypadku słów mu znanych wyświetla podstawową formę wyrazu (ustalaną za pomocą afiksów). W przypadku, gdy nie można rozpoznać słowa, wyświetlane są propozycje najbliższych słów.

- **Analiza morfologiczna**

  Drugim modułem wchodzącym w skład Hunspella jest moduł analizy morfologicznej. Jak napisano wcześniej, sprawdza się on również dla złożonych języków, jak język polski. Poniżej przykładowe wyjście dla tego samego, co powyżej fragmentu:

```
> Wiedza
analyze(Wiedza) =  st:wiedza
stem(Wiedza) = wiedza
> odkryta
analyze(odkryta) =  st:odkryć fl:E
stem(odkryta) = odkryć
> w
analyze(w) =  st:w
stem(w) = w
> procesie
analyze(procesie) =  st:proces fl:Q
stem(procesie) = proces
> dataminingu
Unknown word.
> jest
analyze(jest) =  st:jest
stem(jest) = jest
> wykorzystywana
analyze(wykorzystywana) = st:wykorzystywać fl:E
stem(wykorzystywana) = wykorzystywać
> w
analyze(w) =  st:w
stem(w) = w
> praktyce
analyze(praktyce) =  st:praktyka fl:M
stem(praktyce) = praktyka
> do
analyze(do) =  st:do
stem(do) = do
> analizy
analyze(analizy) =  st:analiza fl:M
stem(analizy) = analiza
> i
analyze(i) =  st:i
stem(i) = i
> poprawy
analyze(poprawy) =  st:poprawa fl:M
```

```
stem(poprawy) = poprawa
> jakości
analyze(jakości) =  st:jakość fl:M
stem(jakości) = jakość
> różnych
analyze(różnych) =  st:różny fl:X
stem(różnych) = różny
> czynników
analyze(czynników) =  st:czynnik fl:T
stem(czynników) = czynnik
```

Tu dodatkowo otrzymujemy informację o wzorcu odmiany każdego słowa.

2.3 Porównanie złożoności i objętości słowników

Jak napisano wcześniej, język polski oraz angielski charakteryzują się diametralnie różną złożonością. Ma to swoje odzwierciedlenie w słownikach.

Tabela 1: Złożoność słowników dla języka polskiego i angielskiego

Cecha	Język polski	Język angielski
Liczba wyrazów w słowniku	280962	46280
Liczba afiksów	7155	1114
Liczba elementów w sekcji REP	64	27

W tabeli 1 zamieszczone są informacje zebrane bezpośrednio z plików słownikowych używanych w programie Hunspell. Jak widać, liczba słów słownika języka angielskiego stanowi tylko 16% liczby słów zawartych w słowniku dla języka polskiego. Podobnie jest w przypadku afiksów - odpowiednio 15,5%.

Doskonałym przykładem będzie ilość możliwych poprawnych form słowa w obu językach. Jako przykład wybrano słowo “pies” (ang. *dog*). Dla języka angielskiego możliwe poprawne formy to: dog, dog’s, dogs, dogs’

Język polski jest pod tym względem znacznie bogatszy. Istnieją w nim poprawne formy: pies, psa, psu, psem, psie, psy, psów, psom, psami, psach. W obu przypadkach ograniczono się tylko do brania pod uwagę podstawowej formy wyrazu (tj. bez zdrobnień, spieszczeń, itp.). Język angielski zawiera tylko cztery poprawne formy, dodatkowo należy zauważyć, że nie różnią się one znacznie od siebie (brak oboczności tematu w odmianie przez przypadki). W odróżnieniu, Polak posługuje się dziesięcioma możliwymi formami słowa “pies”.

Drugim przykładem jest przymiotnik “szybki” (ang. *fast*). W języku angielskim to samo słowo określa również przysłówek “szybko”. Po raz kolejny uproszczono przykład, wzięto pod uwagę tylko formę równą przymiotnika. Język angielski zaskakuje tylko jedną poprawną formą, podczas gdy w języku polskim jest ich aż dwanaście.

Powyższe rozważania jednoznacznie wskazują na zdecydowanie trudniejsze zadanie w przypadku sprawdzania poprawności pisowni tekstu zapisanego w języku polskim w porównaniu z językiem angielskim. Mimo tego faktu jednak, polskie słowniki poprawnej pisowni są tworzone z powodzeniem, w przeciwieństwie do jeszcze bardziej złożonych języków jak na przykład fiński.

3 Możliwości zastosowania

Program Hunspell jest szeroko wykorzystywany w aplikacja GNU. Największy i najbardziej dojrzały pakiet biurowy z tej rodziny oraz najbardziej popularna przeglądarka internetowa korzystają właśnie z niego. W toku badań powstał pomysł, aby za pomocą tej biblioteki dokonywać automatycznej (lub co bardziej prawdopodobne - półautomatycznej) analizy słów kluczowych wybranych dokumentów, a następnie grupowania ich ze względu na słowa kluczowe.

Jeśli chodzi o grupowanie, to taka praca została już wykonana, a jej w pełni funkcjonalna wersja zaprezentowana na obronie licencjackiej autorstwa mojego oraz Tomasza Xięskiego [1, 2].

Do generowania słów kluczowych wykorzystać można analizator morfologiczny Hunspella, który pozwala rozwiązać problem fleksji w języku polskim. Definiując dodatkowo np. słownik wyrazów zabronionych (w którym to znalazłyby się słowa nie wnoszące nic do procesu wyszukiwania), modyfikując słownik tezaurusów tak, aby zawierał również pojęcia dziedzinowe oraz analizując częstość występowania poszczególnych pojęć w analizowanych dokumentach, można zobaczyć zarys automatycznego lub półautomatycznego systemu klasyfikacji.

4 Podsumowanie

W niniejszej pracy starano się przedstawić sposób działania systemu Hunspell. Dokonano również analizy plików słownikowych wykorzystywanych przez ten program oraz przedstawiono różnice pomiędzy złożonością procesu analizy dla języków polskiego i angielskiego.

Przedstawione rozważania mogą posłużyć za wstęp do głębszej analizy problemu i być może rozszerzenia powstałego systemu grupowania tematów prac dyplomowych na podstawie słów kluczowych o automatyczne generowanie tychże w celu zwiększenia wydajności.

Literatura

[1] Jach Tomasz, *Grupowanie jako metoda eksploracji wiedzy w systemach wspomagania decyzji. Analiza algorytmów hierarchicznych.* Sosnowiec, 2008 r.

[2] Xięski Tomasz, *Grupowanie jako metoda eksploracji wiedzy w systemach wspomagania decyzji. Analiza algorytmów niehierarchicznych, k-optymalizacyjnych.* Sosnowiec, 2008 r.

[3] Mario Li Ulfe, *Uzasadnienie pisowni angielskiej*, Wrocław, 1999 r.

[4] Wzorce odmiany dla języka polskiego [w:] *Współczesny słownik języka polskiego*, Warszawa, 1996 r.

[5] László Németh, *Dokumentacja programu Hunspell*, [on-line] `http://sourceforge.net/docman/?group_id=143754`

[6] Portal tłumaczy programu OpenOffice.org `http://lingucomponent.openoffice.org`

[7] `http://pl.wikipedia.org`

[8] `http://en.wikipedia.org`

[9] Pliki man programu Hunspell